Ensemble ML Modelling to classify Real/Fake jobs

By Varad Sunil Jadhav
August Machine Learning Batch

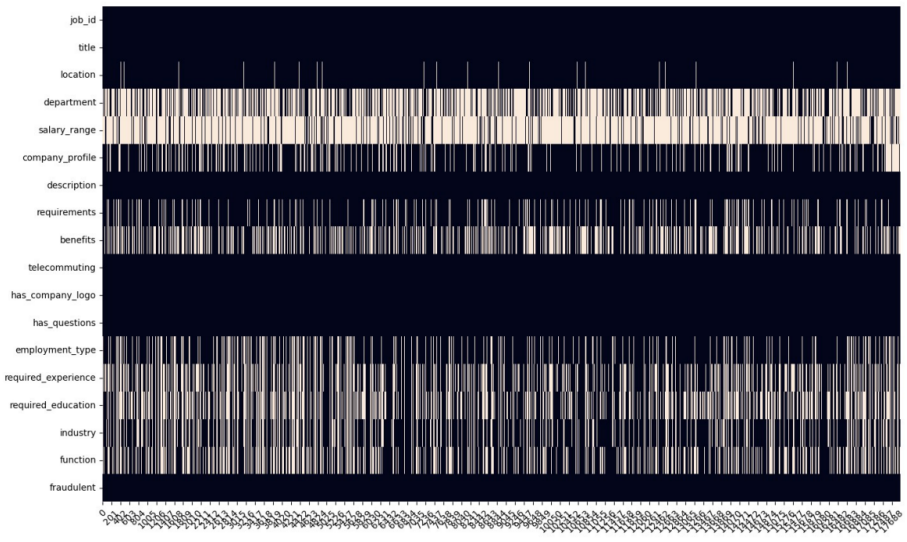
Google Colab Link -
https://colab.research.google.com/drive/1jMV7f7XiVmKwl7n3UfJ21ulBQevDBQgV?usp=sharing

Exploratory data analysis and Data Cleaning

fake_job_postings dataset initially has shape of (17880 * 18) where the features are 'job_id', 'title', 'location', 'department', 'salary_range', 'company_profile', 'description', 'requirements', 'benefits', 'telecommuting', 'has_company_logo', 'has_questions', 'employment_type', 'required_experience', 'required_education', 'industry', 'function', 'fraudulent'.

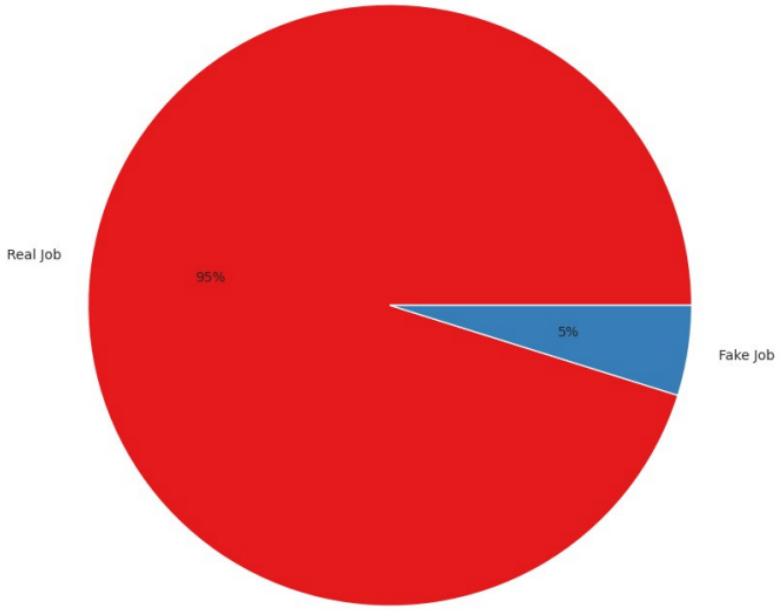
Textual Features	Categorical Features
['title','benefits','company_profile','location','description','requirements']	['employment_type','required_experience','required_education','industry','function','telecommuting','has_company_logo','has_questions']

The heatmap for the dataset.

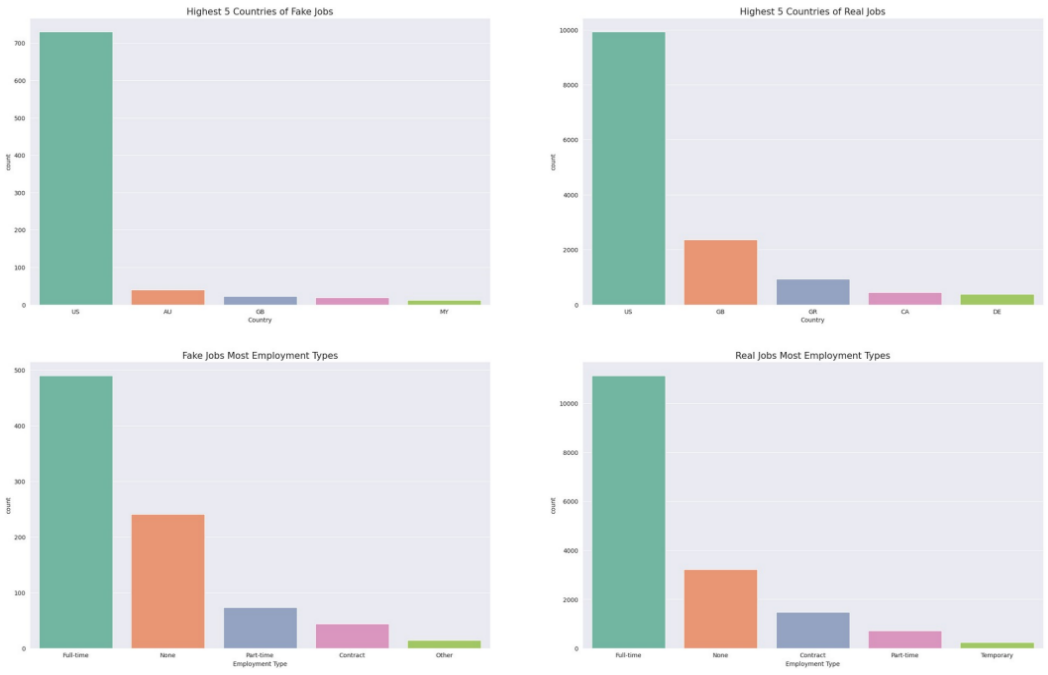


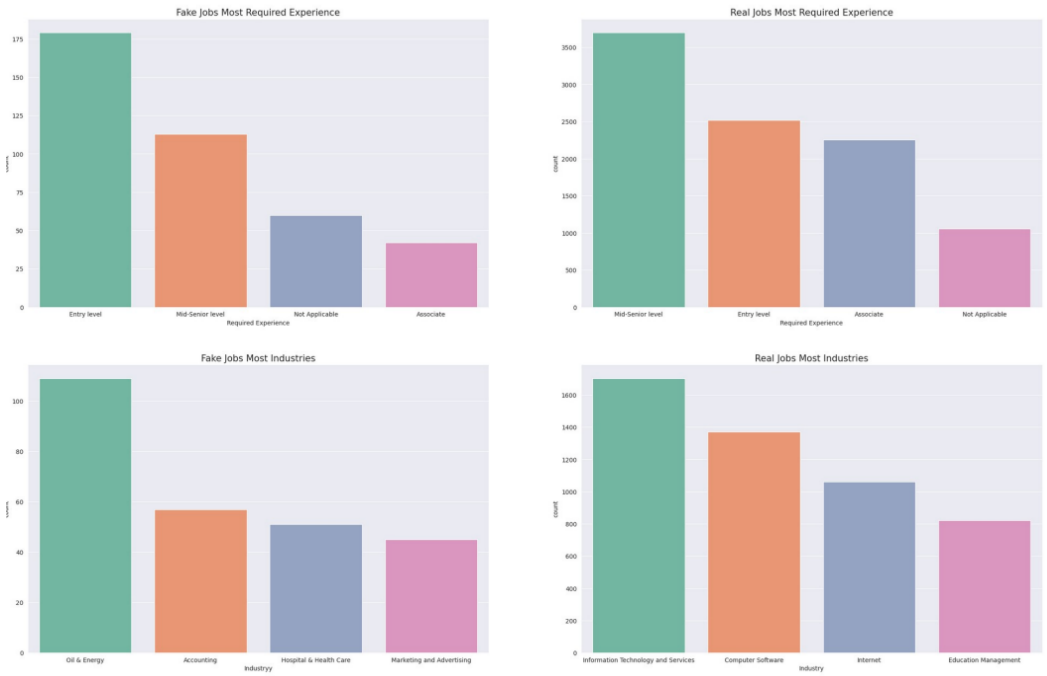
We can see that the features **department** and **salary_range** have much more missing values. We can not remove the **department** feature so merge with the **function** for the missing values. After that I have removed the features **job_id**, **department** and **salary_range**. Other features are important here.

We can see that there is huge imbalance in Dependent variable ‘fraudulent’



Visualising different features





We can get **country** with the help of a feature **location.**

To deal with textual features we have to perform **Natural Language Processing** over each of the textual features using libraries like **nltk** , **wordcloud, etc.**

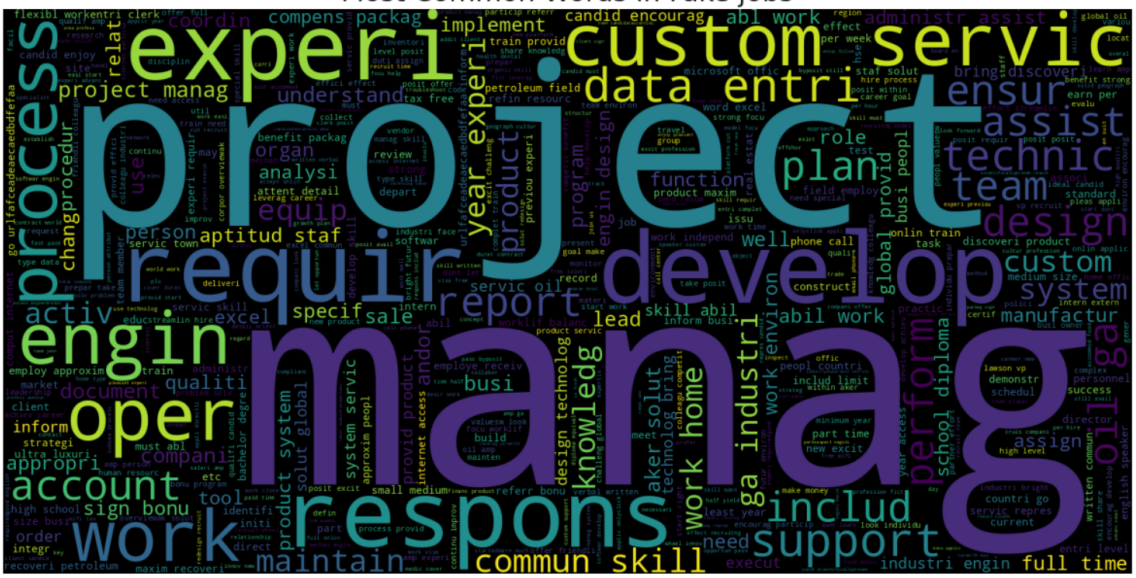
Steps followed for NLP:

1] Removing **stopwords**.

2] **Stemming** remaining words using **Porterstemmer** of nltk

3] Making array of such stemmed words

4] Joining such words with spaces between them [Corpus formation]
5] Use of **CountVectorizer** from sklearn to convert such words into a vector of numbers based on frequency of each word in each textual column member



Some questions regarding the data!

A] Which industry has the most number of fake jobs?

Within 866 fake jobs **oil and Energy** industry have maximum **109** fake jobs in total.

Oil & Energy	109
Accounting	57
Hospital & Health Care	51
Marketing and Advertising	45
Financial Services	35

B] What are the most common title used in jobs in the US?

295 jobs in US have title **English Teacher Abroad**.

English Teacher Abroad	295
Graduates: English Teacher Abroad (Conversational)	144
Customer Service Associate	136
English Teacher Abroad	89
English Teacher Abroad (Conversational)	83

C] Which department or function has high-paying jobs in the UK?

UK means GB [Great Britain].

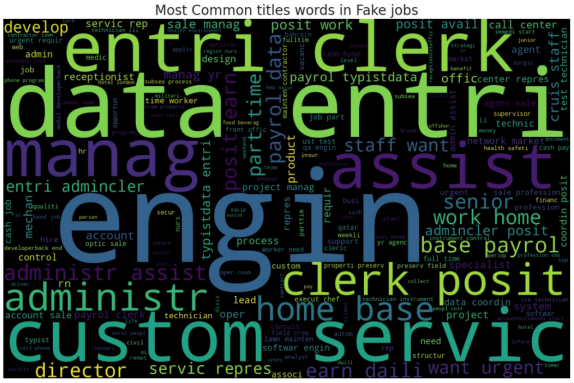
5 most paying departments or functions in UK are-

1) **Financial Analyst** with 112000 $ avg. salary per year
2) **Product Management** with 105000 $ avg. salary per year
3) **Engineering** with 104308 $ avg. salary per year
4) **Administrative** with 102664 $ avg. salary per year
5) **Accounting/Auditing** with 102366 $ avg. salary per year

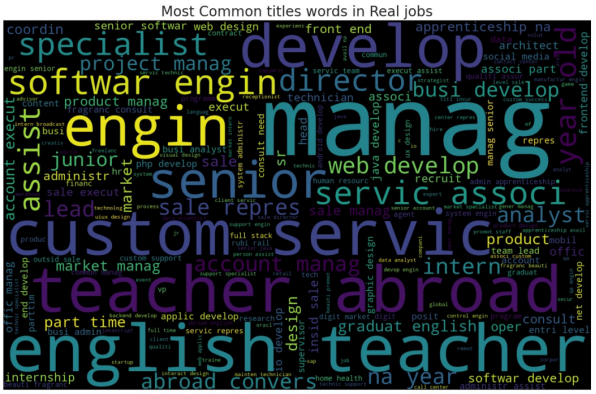
D] What are the commonly used words in textual features for both fake and real jobs?

1) Titles

Most Common titles words in Fake jobs

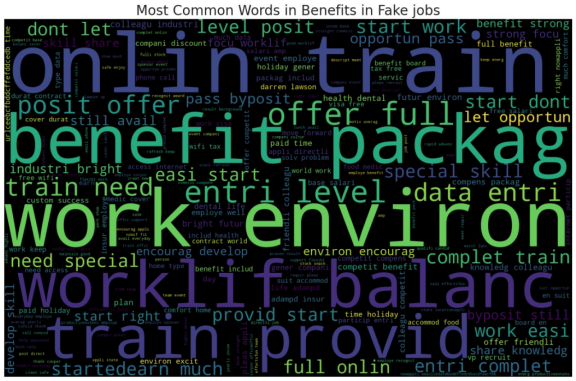


Most Common titles words in Real jobs

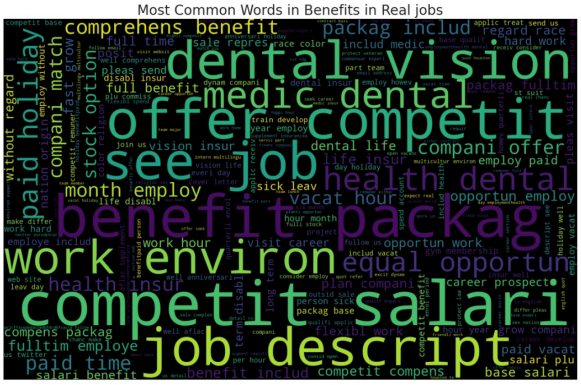


2) Benefits

Most Common Words in Benefits in Fake jobs

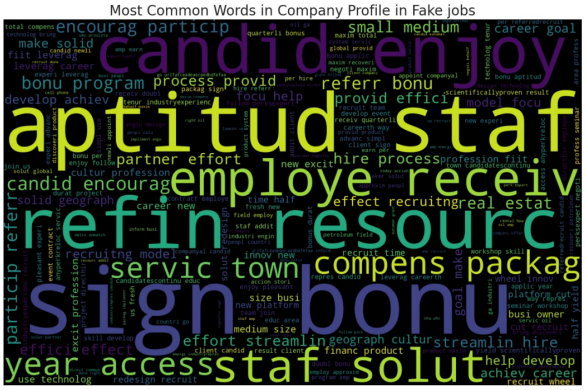


Most Common Words in Benefits in Real jobs



3) Company Profile

Most Common Words in Company Profile in Fake jobs

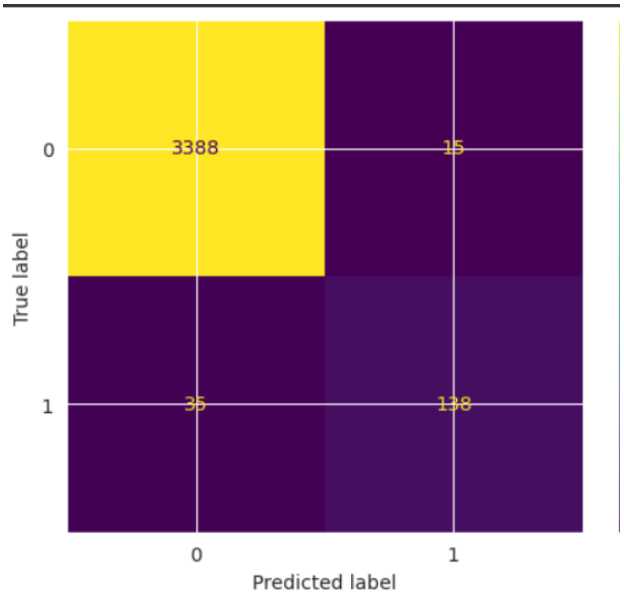


Most Common Words in Company Profile in Real jobs

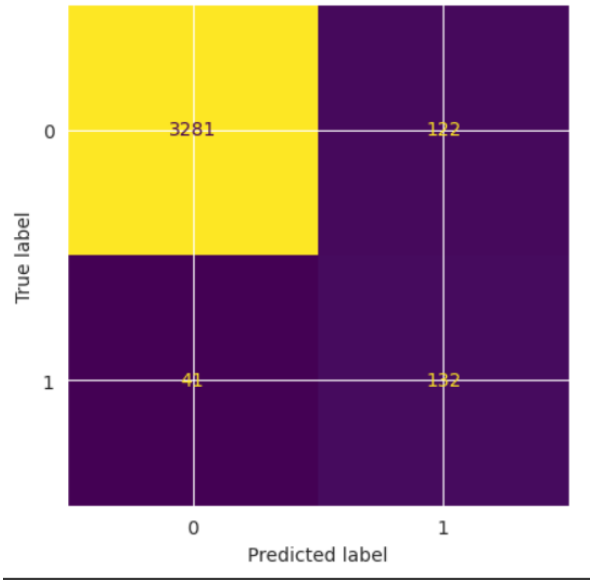
Ensemble Machine learning Modelling

I divided the data into **80:20** train-test-split
Confusion matrix is used because of categorical labels.

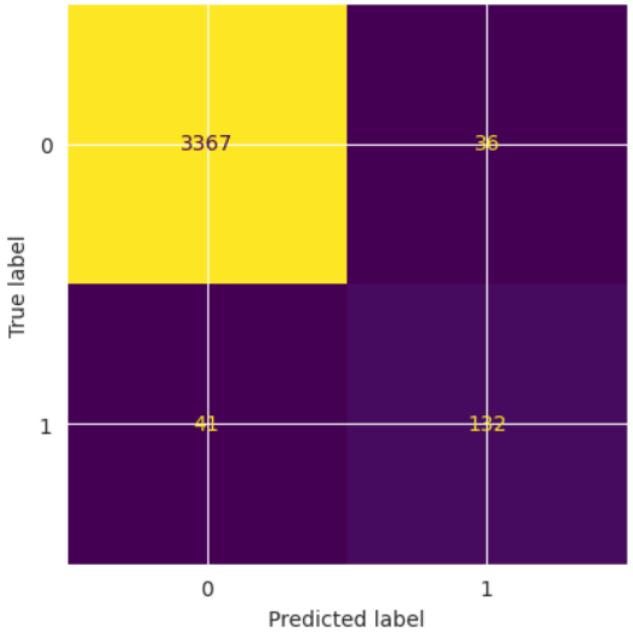
1] **Logistic Regression**
Training data accuracy = 100%
Testing data accuracy = **99%**
Confusing matrix:



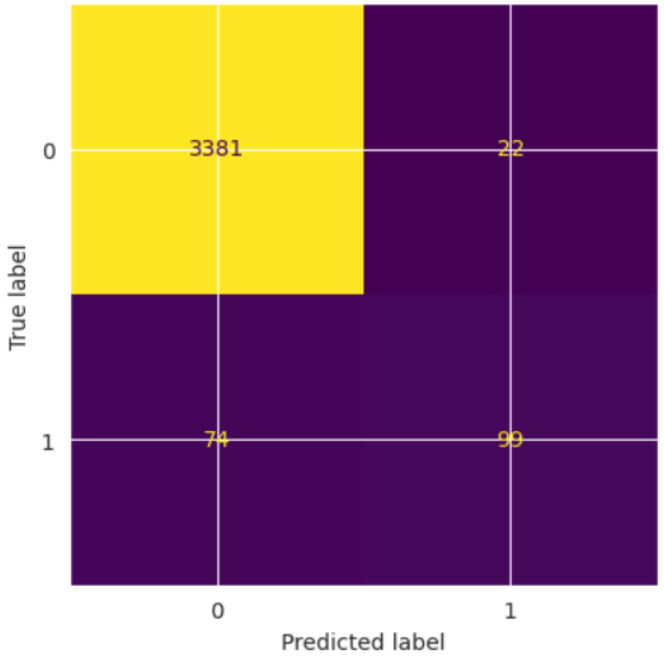
2] **KNN**
Training data accuracy = 96%
Testing data accuracy = **95%**
Confusing matrix:



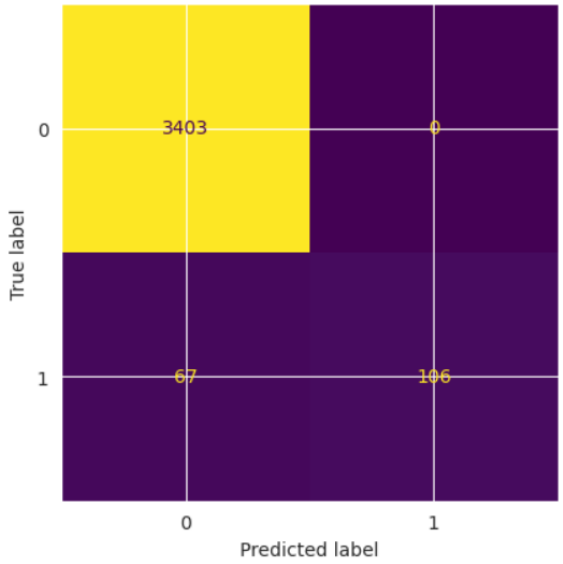
3] **Decision Tree**
Training data accuracy = 100%
Testing data accuracy = **99%**
Confusing matrix:



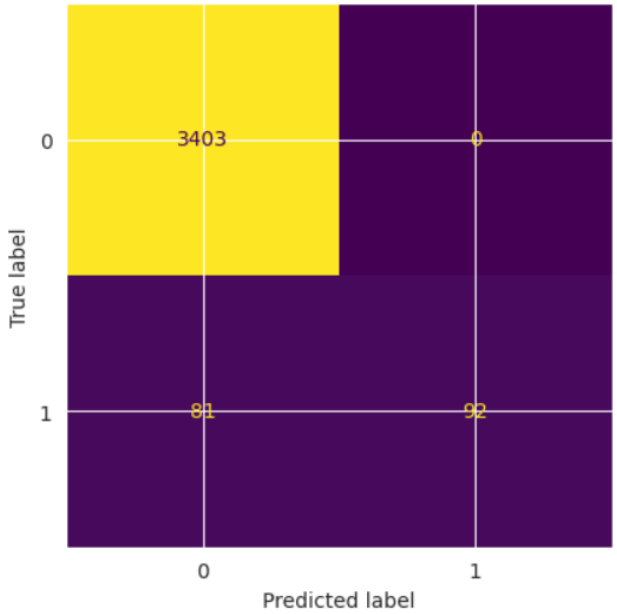
4] **Adaboost**
Training data accuracy = 97%
Testing data accuracy = **97%**
Confusing matrix:



5] **Random Forest**
Training data accuracy = 100%
Testing data accuracy = **98%**
Confusing matrix:



6] **Support Vector Machine**
Training data accuracy = 98%
Testing data accuracy = **98%**
Confusing matrix:



So the best accuracy achieved is using **Logistic Regression** which is 99%.